The Integrity of FPGA Designs: Capabilities Enabled by Unlocking Bitstreams and 3rd-Party IP

Jonathan Graf, Scott Harper, and Lee Lerner
{grafj, harpers, lernerl}@lunainnovations.com
Secure Computing and Communications Group
Luna Innovations Incorporated
1 Riverside Circle, Suite 400, Roanoke, Virginia 24016

Abstract: *We introduce a novel, broad definition of Field Programmable Gate Array design integrity. We claim that an FPGA design with integrity must continuously provide the FPGA user with the function described by the designer and no other function. Using this definition, we explore its value to the domains of FPGA Trust, high-reliability FPGA design, and FPGA design anti-obsolescence. Further, we describe solutions in each of those domains that have a common starting point: Luna's unique software that evaluates the previously inaccessible designs inside of FPGA bitstreams and 3rd-Party Intellectual Property.*

Keywords: FPGA; field programmable gate array; security; integrity; trust; reliability; reverse engineering

Introduction

Every Field Programmable Gate Array (FPGA) embodies its core concept: that its function is realized by a specialized, custom design that was created separately and by different agency than the general design of the silicon of the FPGA. Thus, when considering integrity for these devices, we must address two aspects: the FPGA vendor architecture and the user design. The purpose of the vendor architecture is to create a general sea of unprogrammed logic that can be configured by the user design to realize the user's intended application-specific function. Protecting and measuring the integrity of the vendor silicon architecture involves the same set of challenges addressed in recent projects to trust Application Specific Integrated Circuits (ASICs) and control ASIC supply chain risk [1]. In this paper, we do not treat the integrity of the vendor device itself. Rather, we focus on the integrity of the user design that is embedded in that device. We introduce a novel, broad definition of FPGA design integrity and demonstrate the value of this definition to the concise statement of FPGA security challenges. Finally, we describe several applied methods for guaranteeing that FPGA design integrity is maintained, each of which make use of unique software that evaluates the contents of FPGA bitstreams and 3rd-Party Intellectual Property (3PIP).

Definition and Attributes of FPGA Design Integrity

An FPGA design with *integrity* continuously provides the *User* with the function described by the *Designer* and no other function. The *User* is the party wishing to make use of the function in the FPGA, and the *Designer* is the party or parties responsible for creating the design that realizes that function in the FPGA. The above definition leads to the following three guarantees that must be provided to the User in order for the design to have integrity.

1. A User-trusted description of the function
2. The function described is realized in the design
3. The design realizes only the function described

An FPGA design that cannot guarantee all of the above attributes cannot be said to have complete integrity.[1]

Note that this definition is much broader that the more common use of the term integrity in reference to FPGAs. Traditionally, integrity is used to indicate that a mechanism such as a hash or checksum is used to test the bitstream as it is being loaded on the FPGA device to ensure it has not been changed since it was first generated by the designer. This narrow definition of FPGA integrity falls within our use of the word as one aspect, but it does not comprise the full range of the term's potential. Instead, using the above attributes of an FPGA design with integrity, we may explore how this simple definition elegantly expresses the commonality of the goals within many FPGA security domains. While traversing these topics, we provide examples of technology solutions provided by Luna Innovations that make use of our ability to directly evaluate the designs inside bitstreams and 3PIP. First, however, we make brief comments on the common formats of both the user-trusted description of the function and the design itself.

Functional Description Formats

Before we address design integrity challenges and the technologies used address them, we must briefly consider the functional description formats used to describe the designer's intent. Luna's FPGA integrity technologies are intended to operate with a variety of functional specification formats. As new forms of functional specification are developed, our technology will adapt to accommodate those formats common in FPGA design flows. Depending on the particular case in which we are seeking to guarantee FPGA integrity, we may use the

[1] A similar set of attributes may be easily developed to describe the integrity of any kind of microelectronic design.

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **MAR 2012**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**

4. TITLE AND SUBTITLE
The Integrity of FPGA Designs: Capabilities Enabled by Unlocking Bitstreams and 3rd-Party IP

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Secure Computing and Communications Group Luna Innovations Incorporated 1 Riverside Circle, Suite 400, Roanoke, Virginia 24016

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR’S ACRONYM(S)

11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADB379283. GOMACTech 12: Government Microcircuit Applications and Critical Technology Conference (37th) Spanning the Spectrum: Innovations in Micro-Technologies for System Supremacy. Held in Las Vegas, Nevada on 20-22 March 2012.

14. ABSTRACT
We introduce a novel, broad definition of Field Programmable Gate Array design integrity. We claim that an FPGA design with integrity must continuously provide the FPGA user with the function described by the designer and no other function. Using this definition, we explore its value to the domains of FPGA Trust, high-reliability FPGA design, and FPGA design anti-obsolescence. Further, we describe solutions in each of those domains that have a common starting point: Lunas unique software that evaluates the previously inaccessible designs inside of FPGA bitstreams and 3rd-Party Intellectual Property.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**SAR**	**4**	

Hardware Description Language (HDL) source, a simulatable behavioral model, or even a datasheet as the user-trusted functional description. As other functional descriptions gain industry acceptance, our FPGA integrity technologies are malleable to accommodate them.

Evaluating Bitstreams and 3rd-Party IP

Just as the user-trusted functional description may take many forms, we may similarly wish to guarantee the integrity of the design when it is contained in any of a variety of design formats. For FPGAs, the common formats in which a design might be expressed include HDL, synthesized netlist, and bitstream. Luna's work has primarily focused on the challenge of trust when the design is in the synthesized netlist format or in the bitstream format.

Synthesized Netlist Designs: Designs expressed in this format have been synthesized from the HDL created by the designer. Once synthesized, the design may still be represented in an HDL such as Verilog or in another common electronic design format such as the Electronic Design Interchange Format (EDIF). Whether Verilog or EDIF, however, the synthesized netlist is expressed as a connected and configured arrangement of the FPGA resources necessary to realize the design. Third-Party Intellectual Property (3PIP) cores are commonly distributed to the user purchasing the core as synthesized netlists targeted towards the resources provided by the FPGA of interest. 3PIP core designers commonly encrypt and obfuscate the proprietary implementation details of their cores. These measures taken to hide details of the design add to the challenge of guaranteeing the design's integrity. However, Luna has developed technologies and techniques to explore the implementation of FPGA 3PIP cores sufficiently either to guarantee or to expose problems with their integrity.

Bitstream Designs: The final format for any FPGA design is the bitstream that is used to configure the silicon of the FPGA to accomplish the user's application-specific task. It has long been desirable to evaluate the bitstream directly in order to verify the contents of the design in its deployed form [2]. The challenge faced by FPGA users has been that the bitstream formats are not documented by FPGA vendors sufficiently to allow evaluation of the user designs they contain. To address this challenge, Luna has developed software that analyzes an FPGA bitstream and describes the design it contains as a synthesized human-readable netlist. This capability enables Luna to make uniquely comprehensive claims about the integrity of FPGA designs all the way down to their bitstream implementation on the vendor silicon.

With the variety of functional description and design formats described, we may now look to various FPGA security challenges and how they are viewed through the lens of our FPGA design integrity definition.

FPGA Trust

The challenge of FPGA Trust is that of guaranteeing both integrity attributes (2) and (3) in reference to attribute (1). Taken on its own, attribute (2) – knowledge that the function described is realized in the design – is the traditional challenge of FPGA design verification. With (2) and (3) taken together, we have a definition of the goal of FPGA Trust. Restated, in FPGA Trust we wish to guarantee that the design provides *only* that function described in the user-trusted description *and nothing more*. Luna has done a variety of work in the FPGA Trust domain, primarily through our work on the DARPA Trust and IRIS programs [3, 4]. We may summarize our work on these projects as three major FPGA Trust domains.

Design-to-Source Trust: In this domain, the user and the designer are both trusted. What is not trusted is the design environment in which the design – as expressed as HDL source – has been transformed into its final implementation format. There are a number of factors that may lead to the lack of trust in the design environment. The most common factors are design software with unknown provenance that transforms the HDL in ways hidden from the user and the threat of an insider that may modify the design. When performing this type of integrity evaluation, we wish to treat the design in its final implementation format, the bitstream. Since we trust the designer in this scenario, we use the designer's HDL source as the trusted design description.

Luna has developed software to automate the many steps required to guarantee that the design has maintained its integrity when being transformed from its HDL into its implementation bitstream. The first automated step is the conversion of the bitstream into a synthesized netlist format. Thereafter, our software evaluates the extracted netlist with reference to the HDL source, applying structural, simulation-based, and formal mathematical algorithms to prove integrity or expose differences. Together, these evaluation methods (described in more detail in [4]) provide a formal mathematical guarantee either that the design contained in the bitstream matches the intent, and only the intent, expressed by the designer in their source HDL or that it does not. In the case that it does not, each non-matching feature is exposed for further consideration. Luna has named the software that performs this kind of evaluation the Change Detection Platform.

Netlist-to-Model Trust: As described in the previous section, the common case of evaluating a netlist is when a user is purchasing a design element from a 3PIP vendor. In this case, the vendor commonly provides a model to serve as a simulation reference to the user. The user relies on this model as an accurate representation of the 3PIP when developing their application. This model may not provide the implementation details of the 3PIP core; it may simply replicate the behavior of the core when simulated. The user is able to evaluate this model freely; however, the 3PIP core itself may be an obfuscated synthesized netlist. Luna has developed technology that can create an evaluable netlist from encrypted and obfuscated 3rd-Party IP. In this netlist-to-model instance, the user is trusted, but the designer – the 3PIP vendor – is not. The design portion we wish to trust is the 3PIP core, and the design specification against which we may evaluate the core is the model provided by the vendor. We may once again use the Luna Change Detection Platform to perform this kind of evaluation.

One feature of the Change Detection Platform that is particularly useful in this kind of evaluation is its ability to create mappings. A mapping is a collection of equivalent reference points between the design under test and its trusted reference. While this mapping technology is useful in the bitstream-to-source case, it is especially valuable in the netlist-to-model case due both to the obfuscation the 3PIP vendor may have instituted in the 3PIP core and the fact that the reference model may be an inexact representation of the implementation details of the core that it models. Commercial mapping tools are not designed to support this type of mapping challenge, so Luna has created mapping technologies that solve the problem. While the mapping step is emphasized in the netlist-to-model trust evaluation, each of the change detection steps mentioned in the bitstream-to-source evaluation are also used to prove whether or not differences exist between the netlist and the model.

Netlist-to-Datasheet Trust: There are many cases in which the user may wish to trust a design for which neither source HDL nor a behavioral model may be referenced. For example, it may be that a 3PIP provider does not provide a trustable behavioral model or that a design has been purchased as a bitstream for which there is no accompanying trusted source HDL. It may be that the only trusted reference available to the user is a datasheet describing the function of the design. As mentioned in the previous two sections, Luna has developed technologies that can convert both 3PIP cores and bitstreams into evaluable synthesized netlists. Thus, the remaining challenge is that of comparing a netlist to the trusted datasheet.

Luna is developing a software platform, the Functional Derivation Platform (FDP), to address this challenge. Our approach is multifaceted. From the design netlist, we derive its function using a combination of novel top-down and bottom-up reverse engineering methods. Working top-down, we define the major functions of the designs and their boundaries and then drill into them hierarchically to further define internal functions. From the bottom up, we transform the unstructured netlist into its basic low-level functional constituents and then work up to define the function of groups of low-level functions. The approach is unified and managed in the FDP such that the top-down and bottom-up methodologies converge, completing our understanding of the netlist. We then make use of semi-automated datasheet analysis techniques to turn the datasheet into provable propositions and compare those propositions against the function derived from the netlist. We are in the process of developing multiple techniques and algorithms to automate the top-down, bottom-up, convergence, and analysis processes involved in our approach. Using this software, we will be able to describe the level of similarity or difference between the trusted datasheet and the design contained in the netlist.

High-Reliability FPGA Applications

Design integrity is also a concern in high-reliability FPGA applications. FPGAs used in the aerospace industry, for example, may be subject to strict design assurance standards, such as DO-254. Designers in high-reliability areas must have the assurance that their FPGA bitstream exactly instantiates their intended design. Until now, their only means of verifying the final implementation of their design has been through board-level testing. Similar to our three stated FPGA Trust scenarios, high-reliability designers may be interested in performing bitstream-to-source, netlist-to-model, and netlist-to-datasheet evaluations for a slightly different purpose. The only difference is the agent of change in each domain. In FPGA Trust, the described evaluations are done to determine if a malicious party has changed the design environment, 3PIP core, or application bitstream to add to, remove from, or modify the application. In FPGA high-reliability applications, the evaluation is done to ensure that no mistake by the designer or in the design software has led to an error in its final implementation. Again, the Luna CDP has the ability to verify design integrity down to the bitstream level, uniquely addressing this challenge.

A further agent of change of concern to some high-reliability aerospace applications is the environment in which the FPGA might be deployed. For example, FPGA users that deploy applications in space wish to know that their designs will maintain integrity in spite of Single Event Upsets (SEUs). Because of our unique capability to evaluate the bitstream, Luna is poised to contribute to high-reliability FPGA design integrity by providing a software-only mechanism for investigating the effects of Single Event Upsets on FPGA bitstreams. We envision the creation of a novel fault emulation platform that models SEU effects on the design in software. We anticipate that this work will yield of new understanding of how to lay out FPGA designs such that their critical functions are less susceptible to failure in the face of SEUs that flip bits in the bitstream. In this way, we improve the radiation tolerance of the design, leading to the maintenance of design integrity when deployed in space. A further effect of such a software platform may be the reduction of time spent on ion beams to settle questions of FPGA radiation tolerance. Rather, many experiments now performed on ion beams might be performed virtually using our software models of the effects of bit flips on the design.

Anti-Obsolescence

The final FPGA design integrity challenge we treat is that of anti-obsolescence. In this instance, the user and the design are both trusted. The environment is not considered. Rather, we here wish to guarantee that the design maintains its integrity regardless of the FPGA on which it is instantiated. In anti-obsolescence, we are concerned with the antiquation of the FPGA silicon on which the design was originally instantiated. In many cases, the source HDL for designs on older FPGAs may be lost, but the useful life of the design has not yet expired. In these instances, our ability to convert bitstreams into netlists again demonstrates its value. We may first recover an HDL representation of the FPGA design from the bitstream. Then we can re-synthesize that design for a more modern FPGA. In this manner, we maintain the integrity of the design independent from the FPGA silicon on which it is instantiated. Our Change Detection Platform can perform formal comparisons to ensure that the design does not change between the devices on which it is realized.

Conclusion

We have presented a novel definition of FPGA design integrity, and we have used it as a means of drawing a cohesive line between various disciplines of FPGA security and reliability. This perspective has proven its value to our work creating software with common features that apply across the various described challenges. Critical among these common features are Luna's bitstream and 3rd-Party IP evaluation software.

ACKNOWLEDGMENTS

Luna gratefully acknowledges that the work described in this document was funded in part by the Defense Advanced Research Projects Agency on the TRUST in ICs and IRIS programs, contract numbers HR0011-08-C-0007 and HR0011-11-C-0068, respectively. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. Approved for Public Release, Distribution Unlimited.

References

1. Collins, D., "TRUST in ICs Program Overview," *Proceedings of GOMACTech 2008,* Las Vegas, Nevada, March 19, 2008.
2. Trimberger, S., "Trusted Design in FPGAs," *Proceedings of the 44th Design Automation Conference*, DAC 2007, San Diego, CA, USA, June 4-8, 2007.
3. Graf, J., J. Hallman, and S. Harper, "Trust in the FPGA Supply Chain Using Physically Unclonable Functions," *Proceedings of GOMACTech 2010*, Reno, NV, March 22, 2010.
4. Graf, J., "Change Detection Platform for FPGA Trust," J. Graf, *Proceedings of GOMACTech 2011,* Orlando, FL, March 23, 2011.